Flight Software Roadmap

Jane Marquart, Code 582

Agenda

- Roadmap Objectives, Goals, Principles & Challenges
- Strategic Process
- Flight Software Evolution
- Mission Drivers
- Technology Roadmaps
- Summary

Roadmap Objectives

- Identify the needed technologies to support NASA missions through 2013
- To define a strategy for selecting needed technologies
- Investigate technologies with high, near-term potential

FSW Goals

- Establish & maintain a strategic planning process that supports technology evolution
- Meet customer needs
- Provide high-quality, low-cost products
- Maintain a framework that supports technology advancements and infusion

Guiding Principles

- Maintain balance between research, development and mission infusion
- Maintain core competency
- Accumulate usable artifacts
- Ensure outside participation
- Effectively integrate processes, products, infrastructure, and workforce

Challenges (1 of 2)

- Meet demands for increasing capability from missions
  - Scientific research goals
  - Remote sensing technology
  - Science data processing method
- Stay abreast of promising technology developments
  - Identify those with greatest potential synergy and return on investment

Challenges (2 of 2)

- Advance state-of-art and state-of-the-practice while simultaneously providing software to projects efficiently and effectively
- Minimizing risk of infusing new technologies

Strategic Process

- Identify mission drivers and goals
- Assess regarding most cost effective approach to meeting needs within current software framework
- Review against current state-of-practice to identify gaps
- If gaps exist, examine state-of-the-art
- Nominate specific technology development goals
  - ROI, cost/benefit,quality,risk, priority, customer & developer buy-in
- Software technologies defined and categorized into three areas:
  - Spacecraft Applications
  - Onboard Data Systems
  - Flight Software Development

Flight Software Environment

What is Flight Software?

- For our purposes: Any software running on-board a satellite
- Mission classes – Remote Sensing
  - Upward (Space) or Download (Earth)
  - Stabilization – spin, gravity gradient, 3-axis control
  - Orbit LEO, HEO, GEO, Lagrange Point, Deep Space, elliptical
  - Science or Technology

Paradigm: Monitor & Control

- Commanding both stored (on platform) and "real-time" (off platform)
- Real-time control
- Telemetry and event messages
- Telemetry monitoring and response
- Time Management
- Health and Safety
- Memory/Table Load and Dump
- Startup and Reset from EEPROM

Monitor and Control What?

- Hardware
  - Science Instruments
    - Mechanisms (on/off, open/close, positioning)
    - Detectors (clear, readout, addressing)
- Spacecraft Subsystems
  - Power
  - Thermal
  - Guidance, Navigation and Control
    - Attitude and Position Determination
    - Attitude Control and Propulsion
    - Safe-hold and recovery modes
    - Most demanding real-time control
  - Radio Communication
- Software
  - Applications
  - Hardware interfaces
  - Core functions

Evolution of Space Data Systems

	GRO		XTE		EO-1/MAP		JWST
Processor	**GRO** **1990** NSSC-1 1MHz 64K 18-bit Words RAM	→	**XTE** **1995** 80386 16MHz 1MB RAM 512KB EEPROM 128KB PROM	→	**EO-1/MAP** **2000-01** MIPS R3000 12MHz 32MB RAM 4MB EEPROM	→	**JWST** **2011** Power PC 133 MHz 50MB RAM* 4MB EEPROM* *more off-board
External Bus	**MIL-STD-1553** 1 M bps	→	**SpaceWire** **IEEE1394** 100-400M bps	→	**SpaceLan** 10M-1Gig bps		
Protocols	**TDM** Fixed Major/Minor Frame	→	**CCSDS** Packet Telemetry Telecommand	→	**IP/SCPS**		
Software	**Assembler** → **C**	→	**C++**	→	**Automated Code Generation**		
Software	**Interrupt Driven** **No Operating System**		**Interrupt Driven** **R/T Operating System**		**Interrupt Driven** **RTOS, Software Bus**		**Interrupt & Data Driven** **RTOS, Dynamic Objects**

Functional Evolution

- Complexity moving from ground to flight
  - FSW costs go up
  - Project costs go down
- Example 1 – Slewing the spacecraft
  - From absolute time-tagged commands computed on the ground and uplinked
  - To desired pointing uplinked and all commands computed on board
- Example 2 – Tracking star selection
  - From stars selected on the ground for each pointing
  - To stars selected in flight processor using onboard star catalog

Mission Drivers

Sample Mission Drivers

MISSION	***Needed Operational Capability***	***Enabling Flight Software Technology***	***Technology Readiness Timeframe***
GLAST	Target of opportunity commands (fast response time). Provides Gamma Ray Burst alerts for other missions	Event-based scheduling.	2002
GPM	Auto retransmit of data	IP Protocols. Reliable file Transfer. File management	2004
MMS	Onboard science data processing, storage, compression. Inter-s/c communication, ranging. Autonomous operation.	File management. Onboard scripting. Autonomous fault resolution.	2004
NPOESS	Discrimination and selection of data. Real-time data delivery. High data rate.	Standard network protocols	2005
LISA	Precision onboard constellation control On-orbit data sharing between 3 instrument on separate s/c	Highly precise and specialized GN&C. Inter-s/c communication.	2006

Applying Strategic Process

- List most significant capability needs
- Associate enabling technologies
- Approximate timeframe to support mission, generally 3-4 years before launch
- Grouped into 3 technology areas:
  - Spacecraft Application Technologies
  - Onboard Data System Technologies
  - Flight Software Development Technologies

Spacecraft Applications Technologies (1 of 2)

ADVANCED AUTONOMY

Goal-Driven Operation
- Basic goal engine
- Distributed Goal Decomposition
- Sci vs. H&S Arbitration
- Science Data Synthesis

Event-Based Scheduling
- *ASAP sequence execution*
- *In-Situ Science Event Response*
- Onboard event-based science planning, scheduling and re-scheduling

Fault Resolution
- *Network Management*
- Network Redundancy Management
- Trend Analysis
- Advanced Fault Detection, Determination, Isolation & Recovery

S/C Operation
- Autonomous Navigation
- Event-Driven Slew to Target
- Adaptive Scheduler
- Multi-S/C Collaboration

(Launch)	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
		Aura		Glast		GPM SDO MMS	NPOESS FKSI	LISA ConX JWST	L-BRDF	MagCon GOES-R	

FONT KEY
- *Italics => Existing or In-devel positioned by completion date*
- Non-italics => Technologies for future missions positioned by development start date

Spacecraft Applications Technologies (2 of 2)

Optimized Science Data Handling

- •**Science Feature Extraction**: Pattern Recognition, Feature Identification, Extraction, Analysis, and Reduction
- •**High Fidelity Science Data Fusion/Compression**
- •**Discrimination & Selection of Science Data**: Onboard Filtering to reduce downlink, Tailored data directly to user
- •**Constellation Data Sharing, Processing, Analysis**

Constellation Management

- •***Formation Flying***:
  - *-Relative Attitude Sensing & Control*
  - - Med. & High Precision Relative Navigation
  - - Collision Avoidance
  - -Cooperative Maneuvering
- •**Sensor Web Support**
  - - Distributed Data Processing

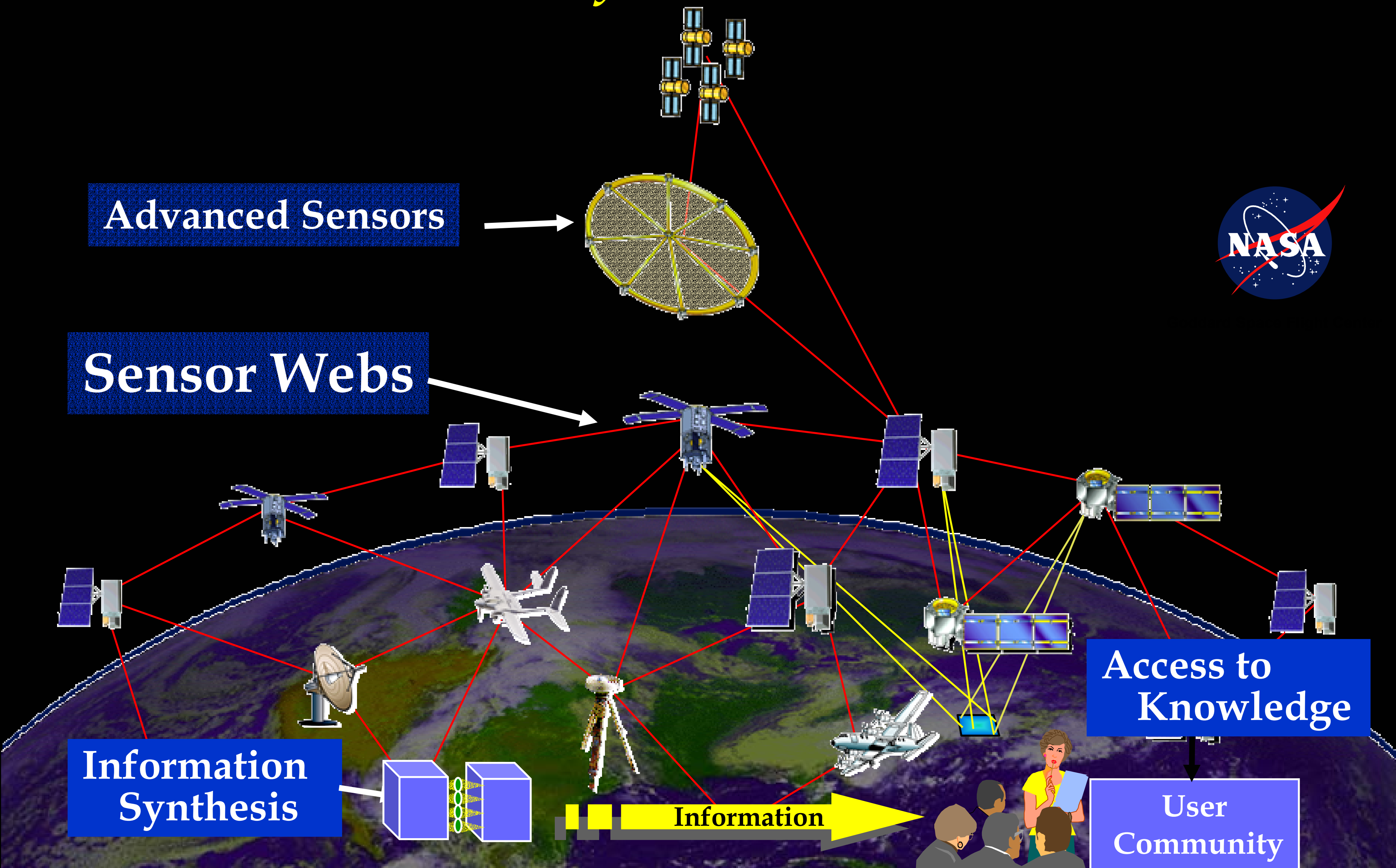
(Launch)	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
		Aura		Glast		GPM SDO MMS	NPOESS FKSI	LISA ConX JWST	L-BRDF	MagCon GOES-R	

FONT KEY

•*Italics =>Existing or In-devel positioned by completion date*

•Non-italics =>Technologies for future missions positioned by development start date

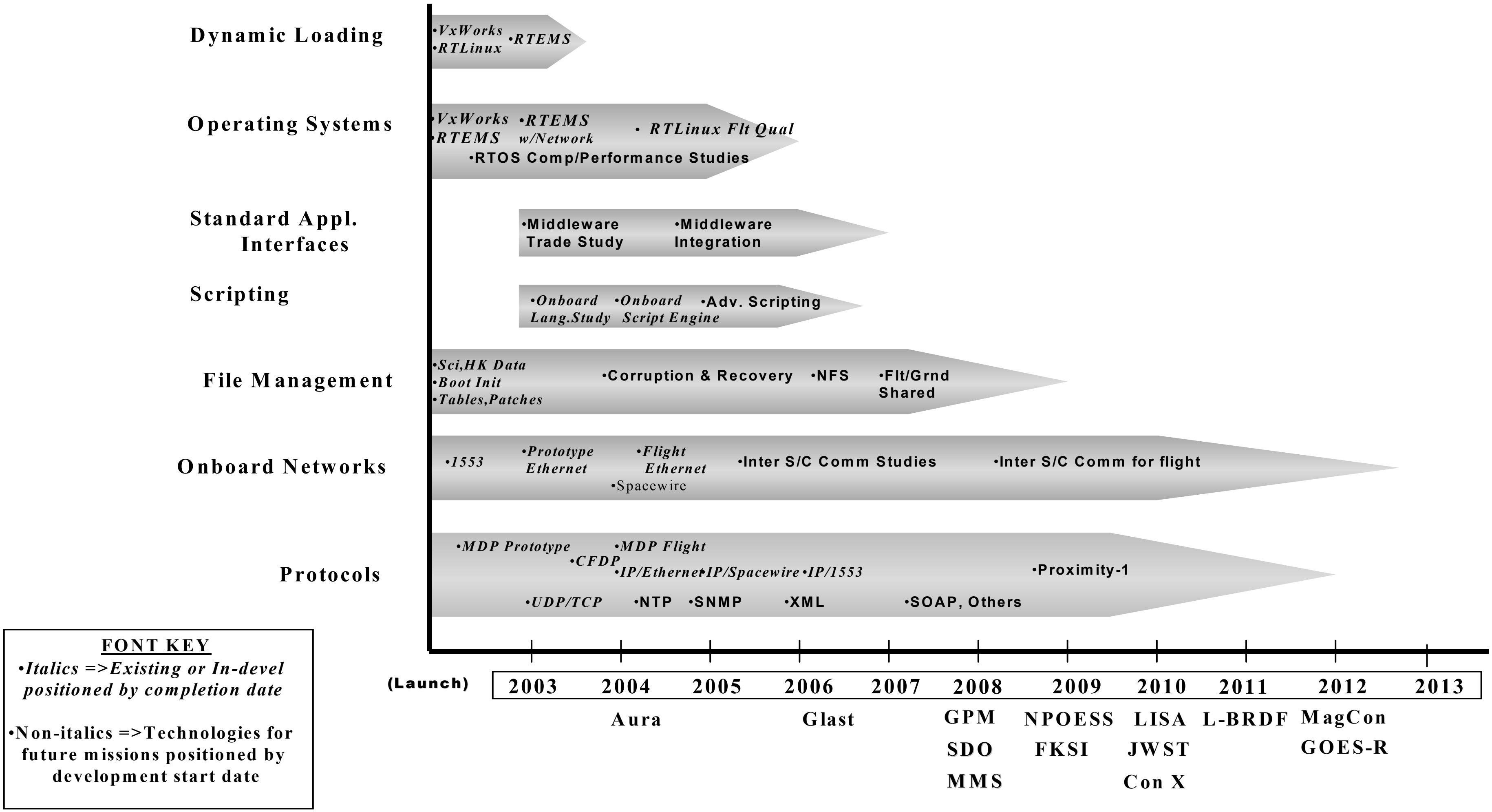
NASA's Earth Science Vision
Architecture of the Future



Technology Capabilities Needed in 2010+ Timeframe to Support Future Science Enterprise Missions

- Multiple spacecraft, multi-instrument, multi-point observations & measurements supporting continuous dynamic studies
  - Constellation management
  - Goal-driven mission control
  - Automated health and status monitoring
  - Triage management- *situation assessment and information synthesis tools to help anomaly diagnosis in "lights out" situations where operators must intervene.*
  - Intelligent user interfaces- *data visualization techniques to reduce information overload and present the results in an easily assimilated manner*.
- Dynamic response to science event detection or changing science priorities
  - Event-responsive control systems
  - Dynamic planning/replanning
  - Goal-driven mission control
  - Spacecraft-initiated communications events/ situation alerts
- Information shared seamlessly between sensors & sciencecraft
  - Common communication schemes
  - Collaborative payload and platform tasking ….

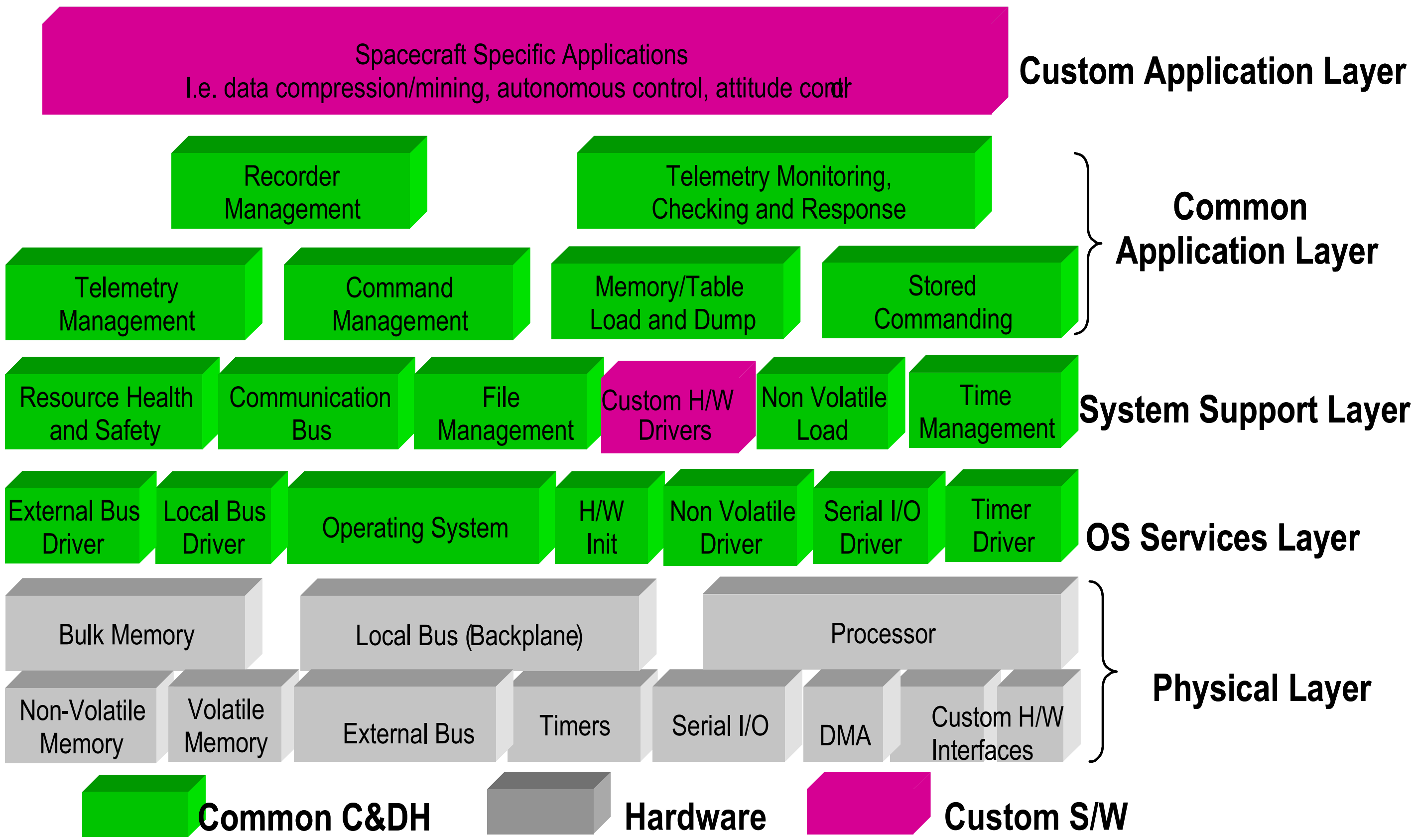
Onboard Data System Technologies



Key Technology: Layered Architecture (1)

- Benefits of layered architecture:
  - Minimizes impact of changes by isolating the extent of the system affected by each change
  - Simplifies configuration
  - Promotes reuse of components
  - Simplifies replacement of components
  - Scalable
- Relevant Activities
  - refinement of architecture using model-based approach, standard interfaces, and automated tools

Key Technology: Layered Architecture (2)



Key Technology: Interface Standards

- Current onboard standards
  - MIL-Std-1553 limits bandwidth
  - CCSDS limits connectivity
- Potential Benefits
  - Improved connectivity
  - Greater flexibility
  - Pluggable components
  - Simpler integration
  - Improved reuse
  - Reduced cost
- Relevant Activities
  - GMSEC
  - SOIF Network Management and Message Service
  - IP prototyping
  - High-speed bus device drivers

Key Technology: File Systems

- Benefits of File Systems
  - Simplifies code, reduces development cost, and improves reuse
  - Greatly improves flexibility of data handling providing exciting features to missions
- Use of File Systems
  - Flight use limited in past by H/W resource constraints
  - Used on Triana (waiting for launch); GPM
  - Potential problems include issues related to data integrity, reliability, and performance
- Possible enhancements to current implementation
  - Corruption and Recovery
  - NFS
  - Flight/Ground file sharing

Key Technology: Linux as an RTOS option

- Why Linux?
  - Has desirable characteristics of an OS: Open source; scalable; stable; has file system support; UNIX compatibility provides a rich and mature set of programming interfaces; supports current network protocols and standards; supports large number of device drivers; well-documented
- Three issues for flight implementation:
  - Can it meet timing requirements for a real-time system?
  - Is it reliable as an embedded Real-time system?
  - Large EEPROM footprint (4Megabytes)
- Update:
  - LynxOS being evaluated for use on current missions

Flight Software Development Technologies

Reuse Library

- Generic Reqts., Repeatable Tests, Standards
- *Informal re-use of C&DH Apps*
  - Generic, reusable C&DH apps
- ***ACS models***

Tools

UML Modeling

- Method & Tools
- CM & Req. Traceability
- Test Scripts
- Performance
- C&DH Models
- Other Subsystem Models

Automated Devel, Testing & Deploy

- Common Reqt Mgmt, CM, DCR Tracking
- Common FSW Test Tools
- Flight/Grnd Database Tools
- Generic Performance Analysis Tools
- Autocode Generation Tools

Unified Simulation & Test Environment

- Easily Configurable Simulation Systems (H/W I/Fs and software)
- Re-usable, Configurable FSW Test Systems Architecture

(Launch)	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
		Aura		Glast		GPM SDO MMS	NPOESS FKSI	LISA JWST Con X	L-BRDF	MagCon GOES-R	

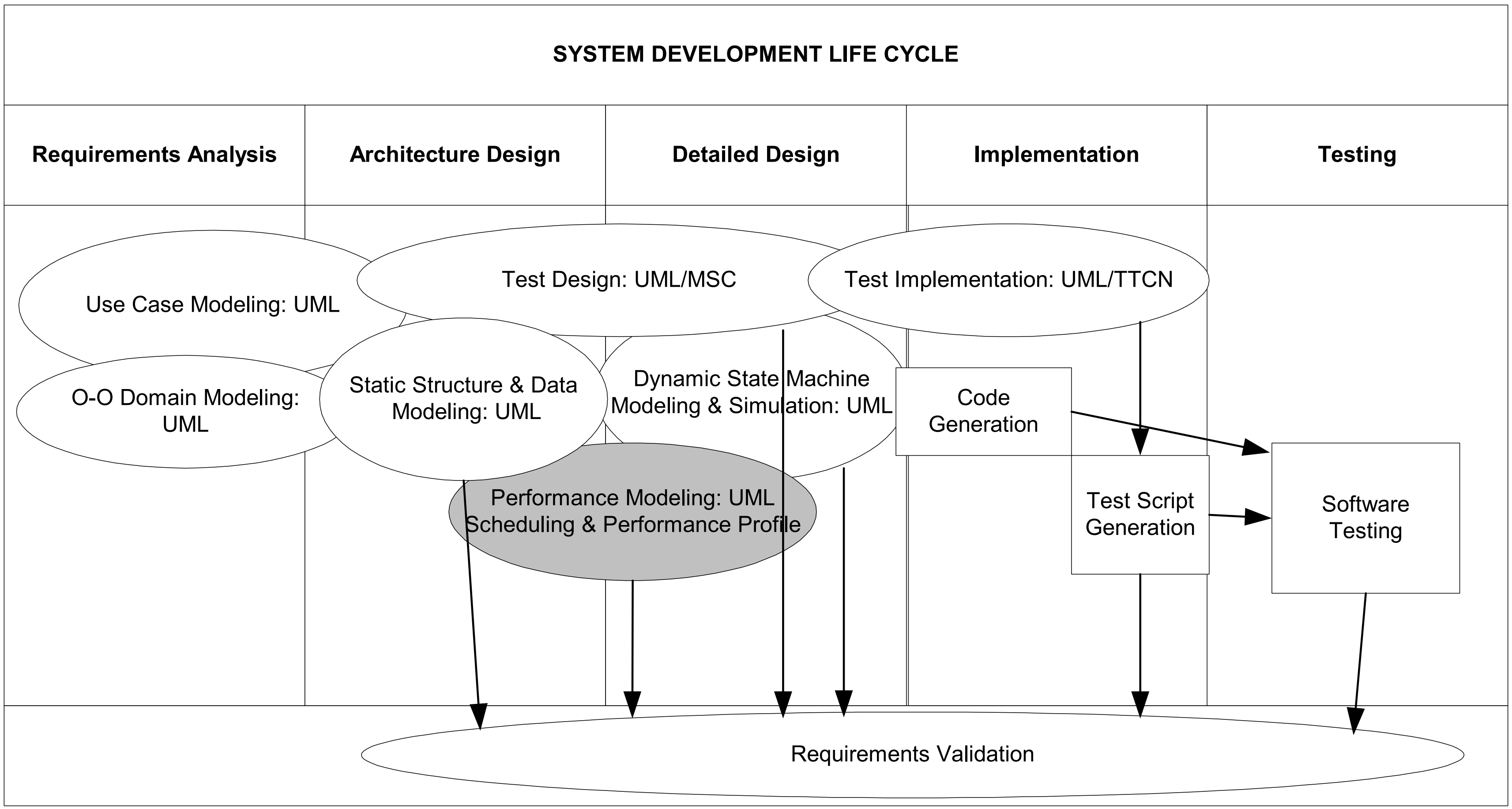
FONT KEY

- *Italics =>Existing or In-devel positioned by completion date*
- Non-italics =>Technologies for future missions positioned by development start date

Key Technology Model Based Development (1)

- Why model based development?
  - Potential for significant time and cost savings in development process
    - Use of models decreases chance of introducing coding errors
    - Simulation validation of design allows for early detection and correction of problems, less costly to fix
    - Use of code and test script generators saves time and cost
- Scope:
  - Select mainstream methodology and commercially available toolset that implements a model-based methodology
  - Develop and test flight software using a model-based approach throughout the entire life cycle
  - Generate a Set of Re-usable Objects for Use on Multiple Future Projects

Key Technology
Model Based Development (2)



Key Technology
Model Based Development (3)

- Results:
  - Dominant software development standards are UML or SDL-based
  - Tool vendors seem to be aligning toward the UML2.0 standard, but no one vendor "has it all"
- Current Use:
  - JWST has been successful
    - Independent simulation models
  - Portability issues between tools

Key Technology: Unified Simulation Environment (1)

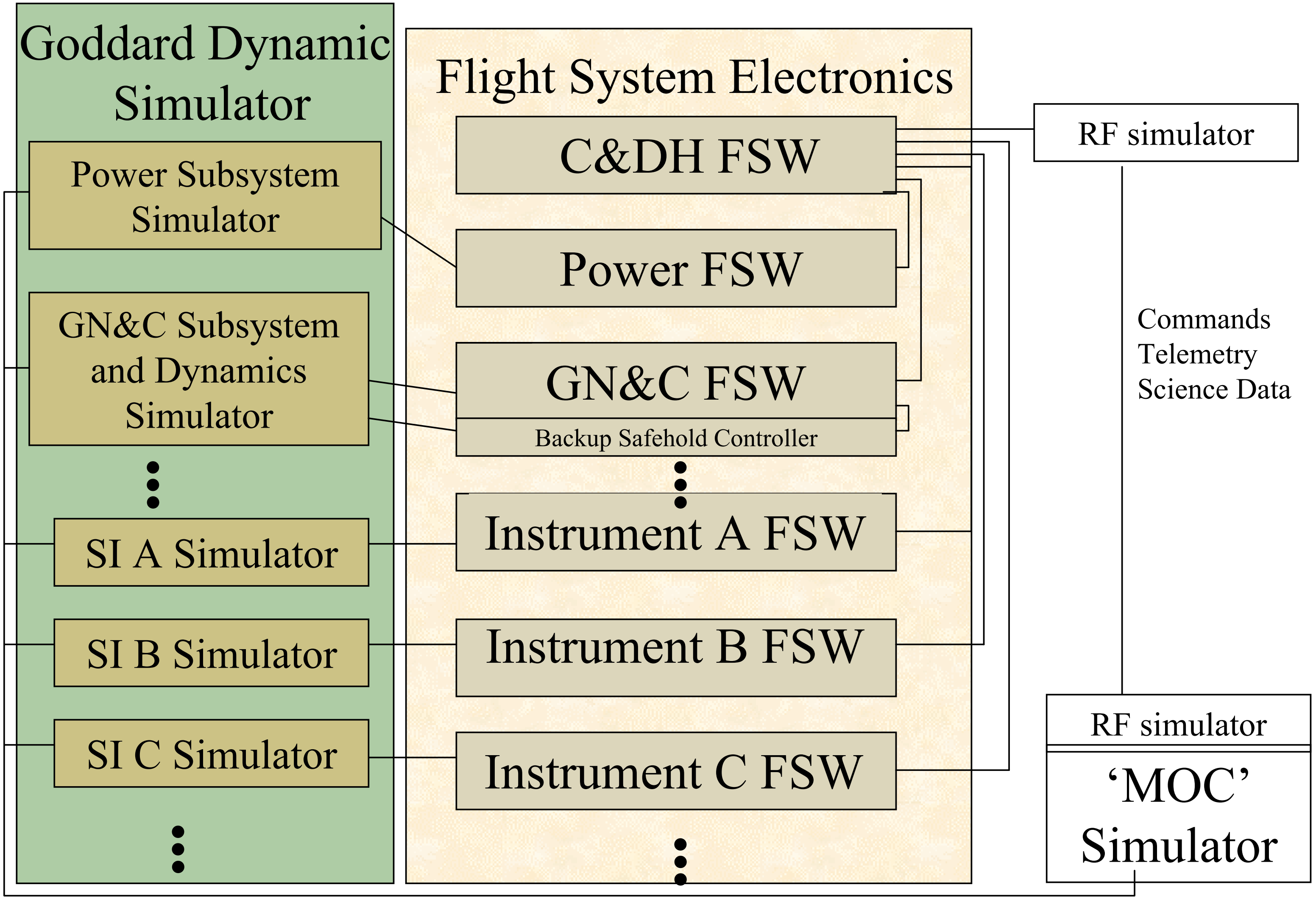
- Why USE?
  - FSW System Acceptance requires high fidelity, on-orbit simulation environment which does not constrain ability to exercise operations and on-orbit contingency scenarios
  - Existing simulation environments cannot accommodate all on-orbit FSW System validation demands.
    - Individual flight box simulators not always tied together for coordination of time, initial conditions, orbital events, spacecraft dynamics, ground communications, etc.
    - ETUs (very expensive) required to test redundancy capabilities
  - Need to improve FSW test productivity

Key Technology: Unified Simulation Environment (2)

- Benefits of a Unified Simulation Environment
  - Standard communication interface between simulators and spacecraft electronics enables coordinated activities resulting in complex mission scenario simulations
  - Central control system provides uniformity among simulators
  - Can provide cheap, easily deployable, pure software simulations of a spacecraft
  - Desktop test capability for developers and testers

- Relevant Activities
  - Goddard Dynamic Simulator (GDS) Proposed and base-lined for SDO and GPM

Key Technology: Unified Simulation Environment (3)



Automated Tools

- Integrated Design and Development Tools
  - Object-Oriented, State-Based Modeling (UML)
  - Automated Code Generation
  - Desktop and target processor deployment
  - Debug facility with probes, traces, animation
- Integrated Project Environment
  - Requirements Management
  - Configuration Management
  - Defect and Change Tracking
  - Test Management

Since FY02

- **Onboard Data System Technologies:**
  - Prototype Onboard IP and ethernet
  - LynxOS candidate for mission
  - Evaluation of CFDP
  - Common Flight Executive
- **Flight Software Development Technologies:**
  - UML modeling process
  - Development of reuse library
  - Automated development and test tools

Summary

- Awareness of future mission needs essential to developing required technologies
  - Mission details such as operational (flight) scenarios were hard to find, making definition of required technologies difficult
- Now have baseline for needed technologies with timeframes
  - **Onboard Data Systems technologies are critical to ENABLING spacecraft applications technologies**
  - **Flight Software Development Technologies are essential to reducing cost and development time, and supporting the demand for flexibility**
- Participation and collaboration within, and outside of, GSFC has enhanced the focus of flight software
- Roadmap will evolve and be refined as technologies advance and the strategic process is used

Goddard Space Flight Center

Mount Etna Erupting

Showing the lava flow (vertical red stripe) and the smoke plume